

4、zigbee 网关红外接收

1. 实验目的

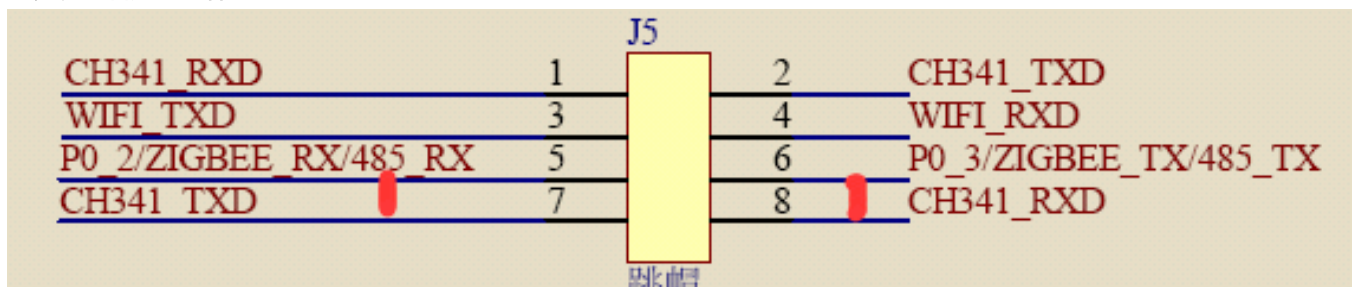
- 1)、通过实验掌握 CC2530 芯片控制红外接收模块 HX1383 的方法
- 2)、掌握 LCD 驱动电路
- 3)、掌握基于红外解码编程

2. 实验设备

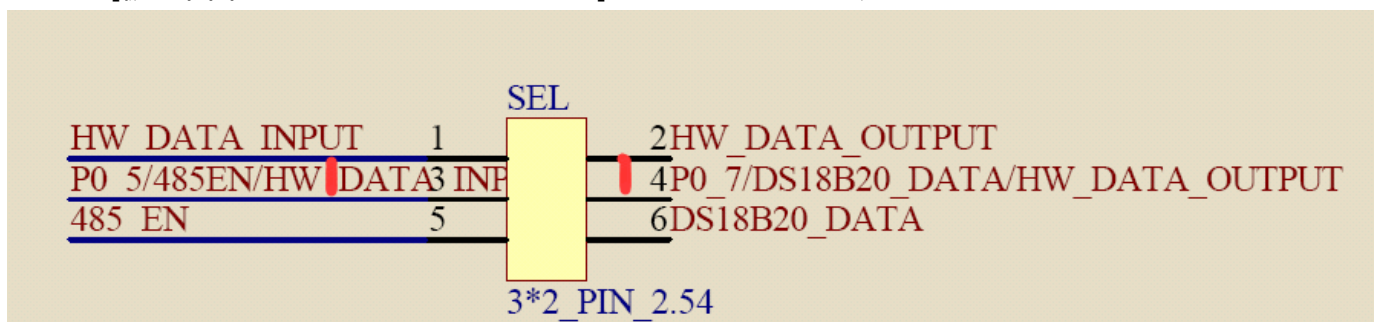
硬件：PC 机一台 ZB2530 网关（底板、核心板、仿真器、USB 线 WIFI 模块）一套

软件：2000/XP/win7 系统，IAR 8.10 集成开发环境

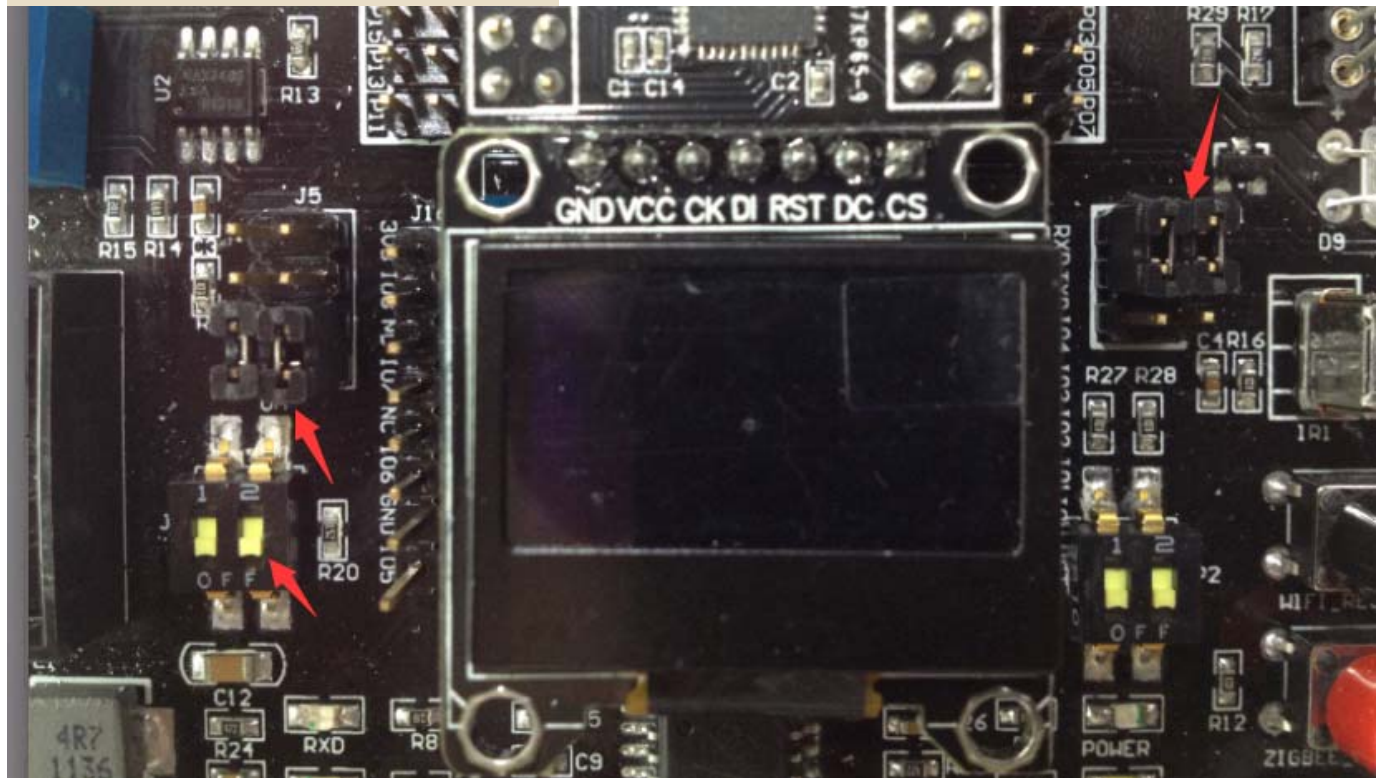
3.实验相关电路图



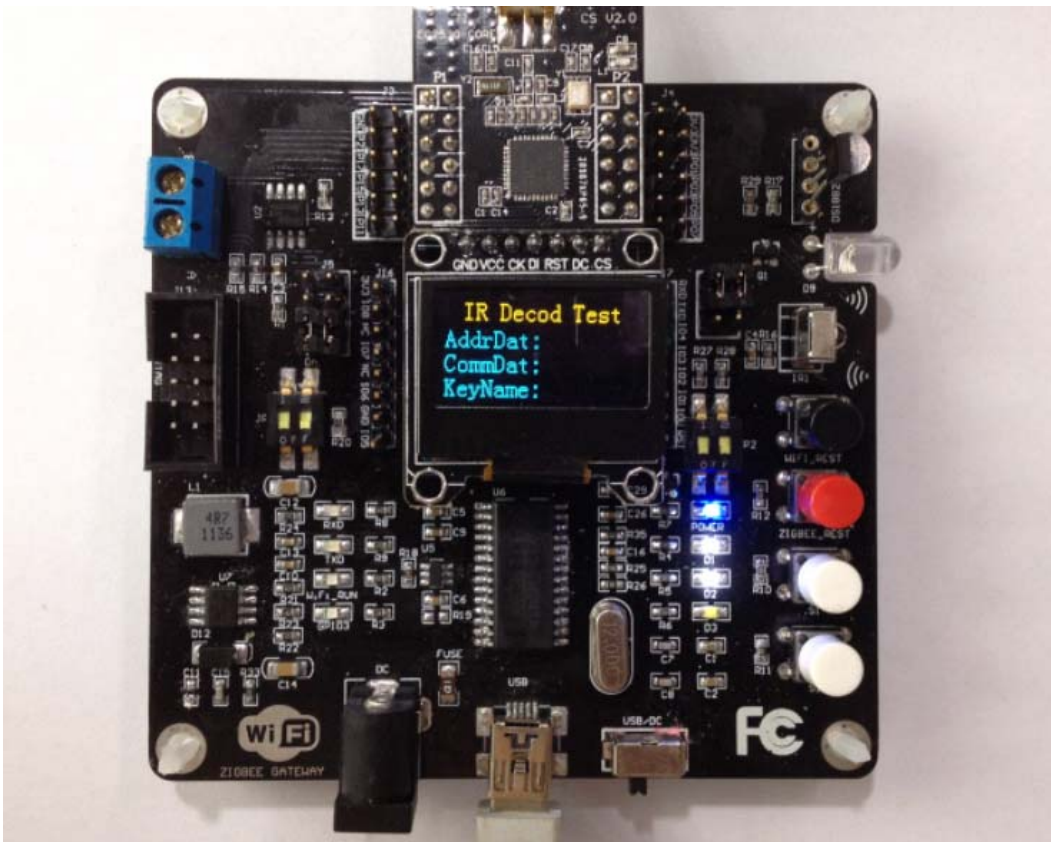
如上图 J5 这样接，5-7 6-8 则表明 zigbee 模块接入到 PC。如果 1-3 2-4 表明 PC 通过 USB 线[板子自带 USB 转串口芯片 CH341]直接接入 WIFI 模块。



如上图 SEL 这样接，1-3 2-4 则表明红外的收发引脚接入到 ZIGBEE 的芯片 IO 进行驱动，其中 HW_DATA_OUTPUT 是红外发射模块。如果 3-5 4-6 表明温度、温湿度和 485 模块直接接入到 ZIGBEE 的芯片 IO 进行驱动。



4. 插好液晶屏



5.硬件初始化

KeyName 就是我们遥控器上面的按键功能

6.代码分析

```
/******
```

* 文件名: IRRecod.c

* 作者: 创思通信

* 修订: 2015-06-08

* 版本: V1.0

* 描述: 红外解码

```
*****/
```

```
/******包含头文件*****
```

```
#include "IRDecod.h"           //红外解码头文件
```

```
#include "uart2.h"             //串口头文件
```

```
#include "lcd12864.h"          //LCD 显示屏头文件
```

```
/******本地变量******/
uint16 HWAddcode,HWComcode;    //红外地址码 操作码
uint16 HWdat[66];              //红外电平长度缓存
/******

* 函数名称 : IRDelay_10us
* 说 明 : 红外解码用延时函数 10 微秒
* 引用说明 : 无
* 返 回 : 无
*****/

void IRDelay_10us(void)
{
    unsigned int a=7;
    while(a)
    {
        a--;
    }
}

/******

* 函数名称 : IRInit
* 说 明 : IR 初始化函数 初始化红外接收头端口
* 引用说明 : 无
* 返 回 : 无
*****/

void IRInit(void)
{
    POSEL &= ~0x20;    //P0_5 设置为普通 IO 口
    PODIR &= ~0x20;    //P0_5 设置为输入模式
    POINP &= ~0x20;    //打开 P0_5 上拉电阻,不影响
}

/******

* 函数名称 : IR_Decoding
```

* 说 明：IR_Decoding 红外解码函数

* 引用说明：先调用 IRInit 初始化

* 返 回：0 解码正确

1 引导码错误

2 用户识别码(地址码)错误

3 操作码（指令码）错误

4 超时出错

*****/

uint8 IR_Decoding(void)

```
{
    uint16 LLevel = 0;        //低电平时长变量
    uint16 HLevel = 0;        //高电平时长变量
    uint8 i=0;                //循环变量
    uint8 AntiCode1 = 0;      //反码验证变量
    uint8 AntiCode2 = 0;      //反码验证变量
    HWAddcode = 0;            //红外地址码变量归零
    HWComcode = 0;            //红外指令码变量归零
    for(i=0;i<33;i++)         //引导码 2bit + 16bit 用户识别码 + 16bit 操作码
    {
        HLevel = 0;          //高电平时长归零
        LLevel = 0;          //低电平时长归零
        while(IR_IRQ==0) //低电平
        {
            LLevel++;        //低电平++
            IRDelay_10us();   //延时 10us
            if(HLevel>1100)return 4; //超时出错
        }

        while(IR_IRQ==1) //高电平
        {
            HLevel++;        //高电平++
```

```
        IRDelay_10us();    //延时 10us
        if(HLevel>1100)return 4; //超时出错
    }
    HWdat[i*2] = LLevel;    //保存低电平时长
    HWdat[(i*2)+1] = HLevel;//保存高电平时长
}
if((HWdat[0]<1100)&&(HWdat[0]>800)&&(HWdat[1]<550)&&(HWdat[1]>350))//
引导码识别 9ms + 4.5ms
{
    for(i=0;i<16;i++)//16bit 用户识别码(地址码)识别
    {

        if((HWdat[i*2+2]<70)&&(HWdat[i*2+2]>40)&&(HWdat[(i*2)+1+2]<70)&&(HWdat
[(i*2)+1+2]>40))//识别逻辑 0
        {
        }
        else
        if((HWdat[i*2+2]<70)&&(HWdat[i*2+2]>40)&&(HWdat[(i*2)+1+2]<195)&&(HWdat[(i*2)
+1+2]>140))//识别逻辑 1
        {
            HWAddcode = HWAddcode | (1<<i);//逻辑 1 组成两个字节
        }
        else return 2;//用户识别码(地址码)错误
    }
    AntiCode1 = (HWAddcode>>8)&0xff;//取出高八位
    if(~AntiCode1!=HWAddcode&0xff)//反码验证
    {
        return 2;//用户识别码(地址码)反码验证错误
    }
    for(i=0;i<16;i++)//16bit 操作码 ( 指令码 ) 识别
    {
```

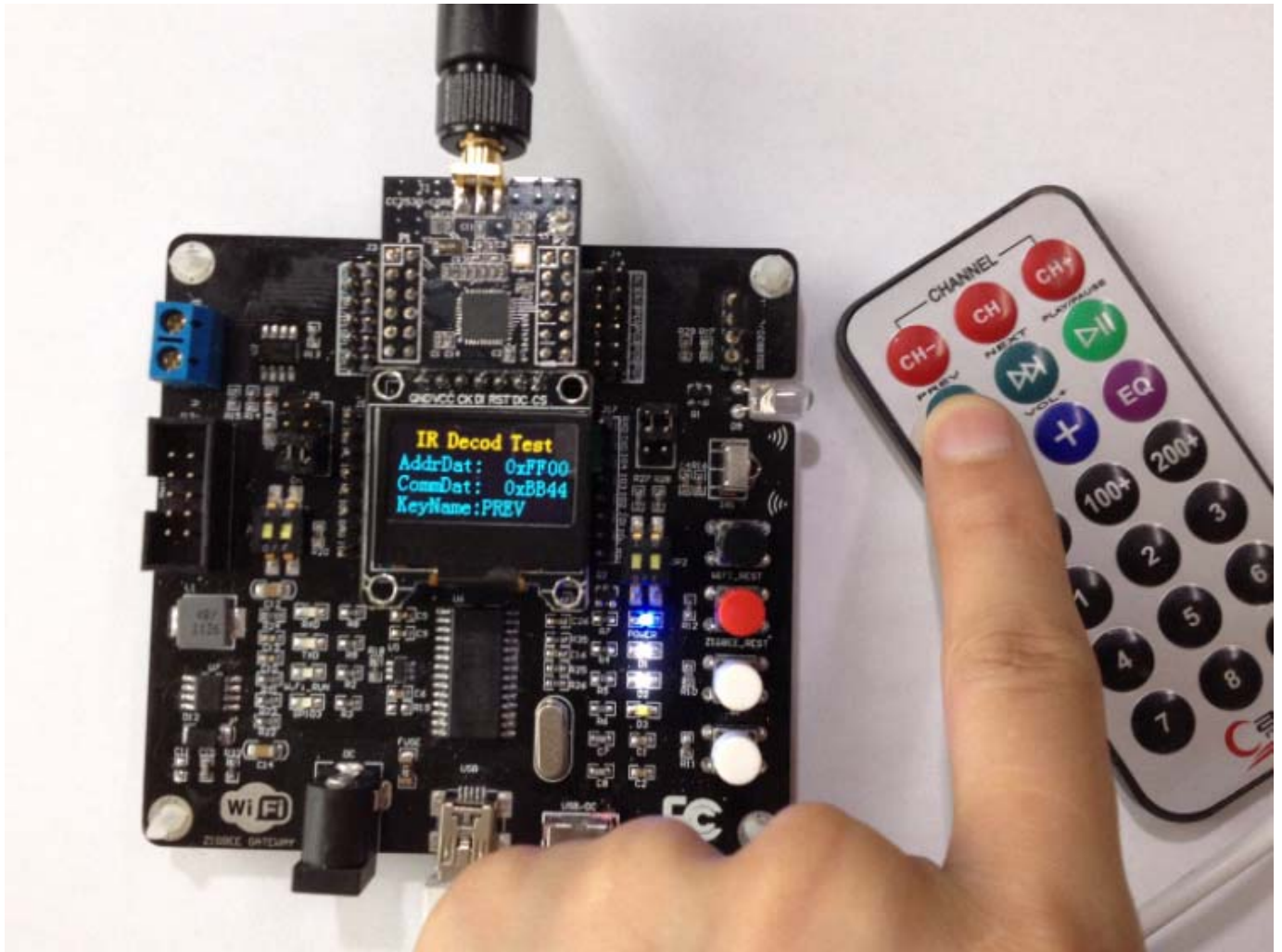


```
if((HWdat[i*2+34]<70)&&(HWdat[i*2+34]>40)&&(HWdat[(i*2)+1+34]<70)&&(HWdat[(i*2)+1+34]>40))//识别逻辑 0
{
}
else
if((HWdat[i*2+34]<70)&&(HWdat[i*2+34]>40)&&(HWdat[(i*2)+1+34]<195)&&(HWdat[(i*2)+1+34]>140))//识别逻辑 1
{
    HWComcode = HWComcode | (1<<i);//逻辑 1 组成两个字节
}
else return 3;//操作码 ( 指令码 ) 错误
}
AntiCode2 = (HWComcode>>8)&0xff;//取出高八位
AntiCode2=~AntiCode2;
if(AntiCode2!=(uint8)(HWComcode&0x00ff))//反码验证
{
    return 5;//用操作码 ( 指令码 ) 反码验证错误
}
}else return 1;//引导码错误
return 0;//解码正确
}
```

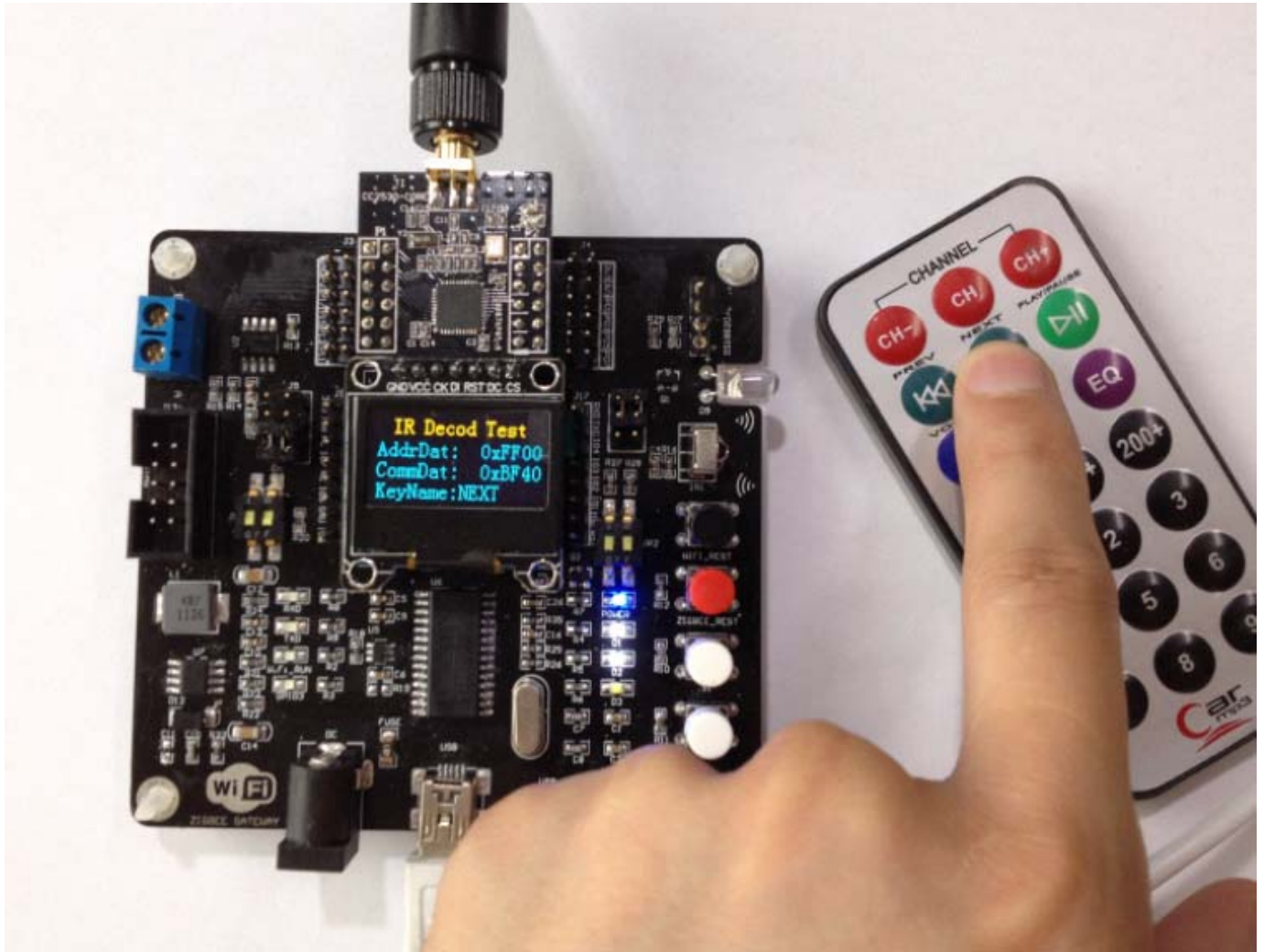
6.实验现象

当把程序下到开发板后，请观察串口信息输出。如果有 LCD，请同时注意 LCD

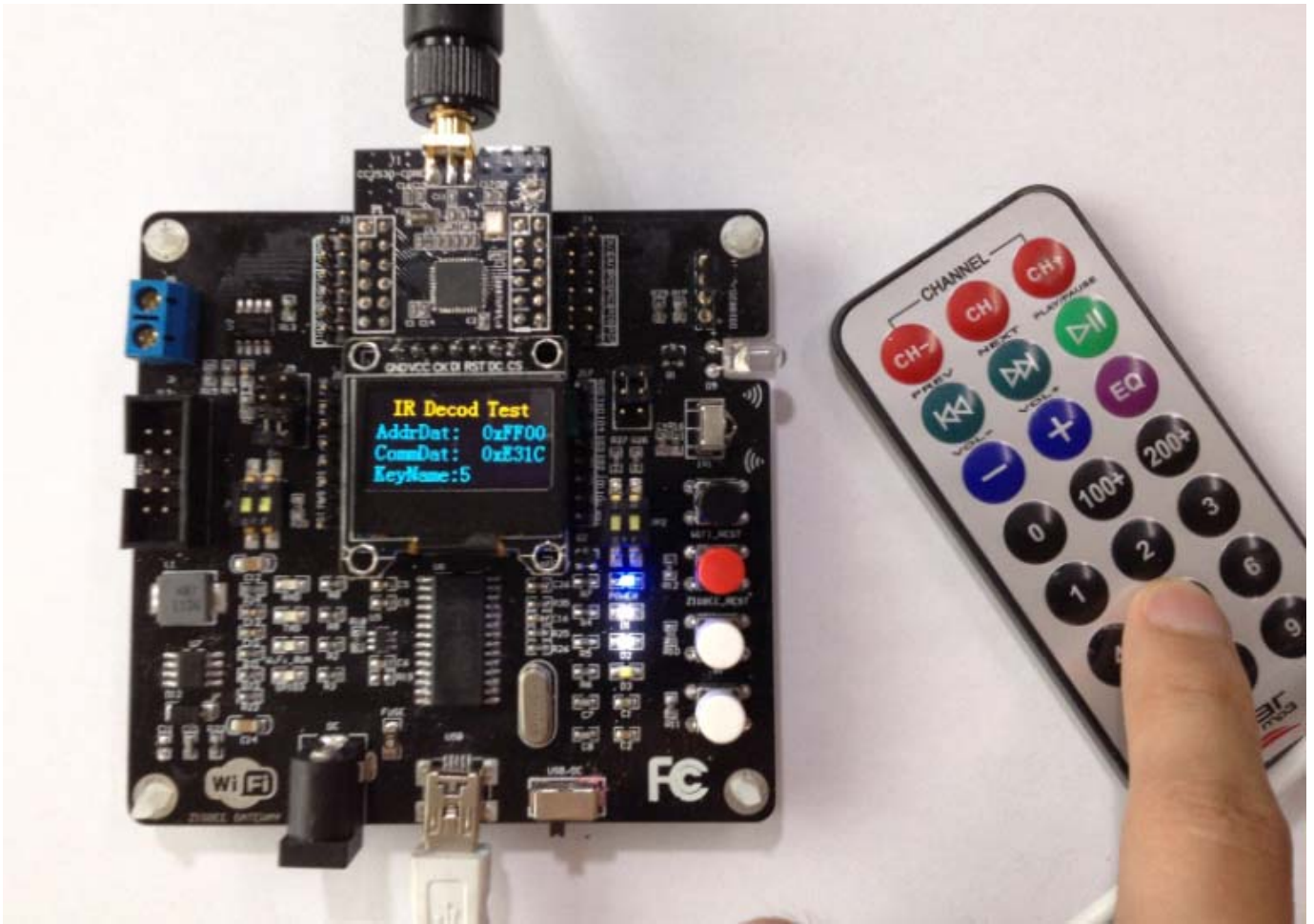
A.按遥控器 PREV。



B.按遥控器 NEXT



C.按遥控器 5



如果没液晶屏的朋友，可以看下串口信息输出；

